AMENDMENTS TO THE CLAIMS

Upon entry of this amendment, the following listing of claims will replace all prior versions and listings of claims in the pending application.

IN THE CLAIMS

Please amend claims 1, 12, 13, 14, 16-22, 25, 30-31, 33-37, 48-50, and 52-58 as follows:

1. (Currently Amended) In an electronic device, a ~~A~~ method for maintaining a database of data objects, comprising:

receiving a request to store in a database capable of storing data objects in a second programming language a first data object implemented in a first programming language including attributes and attribute values for a class;

transforming the first data object to a second data object implemented in a second programming language, wherein the second data object includes the attributes and attribute values of the class included in the first data object; and

~~adding~~ storing, in response to the request, the second data object to the database. ~~wherein the database is capable of storing multiple data objects implemented in the second programming language.~~

2. (Previously Presented) The method of claim 1, further comprising:

receiving a class schema including information on the class and attributes of the first data object; and

using the received class schema to transform the first data object to the second data object.

3. (Previously Presented) The method of claim 2, wherein using the received class schema to transform the first data object to the second data object further comprises:

generating a source code file in the second programming language to implement the class and attributes included in the class schema;

compiling the source code file to generate an executable file that implements methods of the class;

using one method of the class to construct the second data object; and

including the attribute values from the first data object into the second data object.

4. (Previously Presented) The method of claim 3, wherein generating the source code file in the second programming language further comprises:

generating statements into the source code file to define SET and GET interfaces for each attribute in the class.

5. (Previously Presented)The method of claim 4, wherein including the attribute values from the first data object into the second data object further comprises:

using at least one GET method in the first programming language to access the attribute values from the first data object; and

using at least one SET method in the second programming language to set each attribute in the second data object to the corresponding accessed attribute value.

6. (Previously Presented)The method of claim 4, wherein including the attribute values from the first data object into the second data object further comprises:

using at least one GET method in the first programming language to access the attribute values from the first data object; and

generating statements into the source code file to set the attributes in the second data object to the accessed attribute values from the first data object, wherein compiling the source code file produces the second data object with the attribute values set to the attribute values accessed from the first data object.

7. (Previously Presented) The method of claim 2, wherein the class schema includes for each attribute a name, data type and length of the attribute.

8. (Previously Presented) The method of claim 2, wherein the class schema is implemented in an Extensible Markup Language (XML) file.

9. (Previously Presented) The method of claim 1, wherein the database comprises an object oriented database.

10. (Previously Presented) The method of claim 1, wherein the first programming language comprises a non-Java object oriented language and wherein the second programming language comprises the Java programming language.

11. (Previously Presented) The method of claim 1, further comprising:

receiving a third data object implemented in the second programming language; and

adding the third data object to the database.

12. (Currently Amended) The method of claim 1, further comprising:

receiving a request to store in the database a third data object implemented in a third programming language including attributes and attribute values for one class;

transforming the third data object to a fourth data object implemented in the second programming language, wherein the fourth data object includes the attributes and attribute values of the class included in the third data object; and

~~adding~~ storing, in response to the request, the fourth data object to the database.

13. (Currently Amended) In an electronic device, a ~~A~~ method for returning data objects from a database to an application that processes data objects in a first programming language, comprising:

receiving a request from the application for ~~at least one~~ a data object in the database storing the data object in a second programming language, the data object having attributes and attribute values of a class;

accessing ~~each~~ the requested data object from the database, ~~wherein data objects in the database are implemented in a second programming language~~;

transforming ~~each~~ the accessed data object to ~~one~~ a transformed data object implemented in the first programming language, wherein ~~each~~ the transformed data object includes the attributes and attribute values of the class ~~in each~~ of the accessed data object; and

returning, in response to the request, ~~each~~ the transformed data object ~~in the first programming language~~ to the application ~~that initiated the request~~.

14. (Currently Amended) The method of claim 13, wherein transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object further comprises for ~~each~~ the requested data object:

using a GET interface in the second programming language to access the attribute values in the accessed data object; and

using a SET interface in the first programming language to add each accessed attribute value from the accessed data object to the transformed data object.

15. (Previously Presented) The method of claim 13, wherein the application that processes data objects in the first programming language comprises a first application, further comprising:

receiving a request for at least one data object in the database from a second application that processes data objects in the second programming language;

accessing each requested data object from the database; and

returning each data object accessed from the database in response to the request from the second application to the second application.

16. (Currently Amended) The method of claim 13, further comprising:

providing ~~at least one~~ a class schema, wherein ~~each~~ the class schema includes information on one class and attributes of the class of ~~at least one~~ the data object in the database, wherein transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object further comprises, for ~~each~~ the accessed data object, using information on the attributes in the class schema for the class of the accessed data object to transform the accessed data object to the transformed data object.

17. (Currently Amended) The method of claim 16, wherein ~~each~~ the class schema includes a length of each attribute in the class, and wherein using the information on the attributes in the class schema to transform ~~each~~ the accessed data object to ~~one~~ the transformed data object further comprises:

accessing information on the length for each attribute in the class schema to generate the transformed data object ~~to have~~ with a size at least equal to ~~the~~ a sum of lengths of all of the attributes in the class.

18. (Currently Amended) The method of claim 13, wherein the application requesting the ~~at least one~~ data object is capable of processing data objects in one of the first programming language or a third programming language, further comprising:

determining whether the application requesting the ~~at least one~~ data object processes data objects in the first programming language or the third programming language, wherein the step of transforming ~~each~~ the accessed data object to the transformed data object implemented in the first programming language occurs if the application requesting the ~~at least one~~ data object processes data objects in the first programming language;

transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object implemented in the third programming language if the application requesting the ~~at least one~~ data object processes data objects in the third programming language; and

returning ~~each~~ the transformed data object in the third programming language to the application that initiated the request.

19. (Currently Amended) In an electronic device, a ~~A~~ method for providing information on a class, comprising:

receiving a definition of a class and attributes in the class of a first data object implemented in a first programming language;

generating a file; and

adding information to the file to provide a class schema representing ~~on~~ the class and each attribute in the class. ~~received class definition to the generated file.~~

20. (Currently Amended) The method of claim 19, comprising ~~wherein adding information on each attribute to the generated file further comprises:~~

~~adding information on a name, length and data type of each attribute in the class in the received class definition to the generated file.~~

providing the file with a request to take an action on the first data object, the action associated with a second programming language; and

establishing, using the class schema from the file, a second data object implemented in the second programming language to represent the first data object.

21. (Currently Amended) The method of claim 19, further comprising:

generating at least one tagged element into the file ~~including information on each~~ to represent at least one attribute in the class.

22. (Currently Amended) The method of claim 21, wherein generating the at least one tagged element into the file for each attribute in the class further comprises for each attribute of the class:

generating one tagged element into the file including information on one of a name, a length, and a data type of the attribute. ~~[[;]]~~

~~generating one tagged element into the file including information on a length of the attribute; and~~

~~generating one tagged element into the file including information on a data type of the attribute.~~

23. (Previously Presented) The method of claim 21, wherein the generated file comprises an Extensible Markup Language (XML) file.

24. (Previously Presented) The method of claim 19, further comprising:

accessing the definition of the class, including information on attributes of the class, from a source code file of the class.

25. (Currently Amended) A system for maintaining a database of data objects, comprising: a computer readable medium including the database of data objects;

means for receiving a request to store in a database capable of storing data objects in a second programming language a first data object implemented in a first programming language including attributes and attribute values for a class;

means for transforming the first data object to a second data object implemented in a second programming language, wherein the second data object includes the attributes and attribute values of the class included in the first data object; and

means for ~~adding~~ storing, in response to the request, the second data object to the database~~, wherein the database stores data objects implemented in the second programming language~~.

26. (Previously Presented) The system of claim 25, further comprising:

means for receiving a class schema including information on the class and attributes of the first data object; and

means for using the received class schema to transform the first data object to the second data object.

27. (Previously Presented) The system of claim 26, wherein the means for using the received class schema to transform the first data object to the second data object further performs:

generating a source code file in the second programming language to implement the class and attributes included in the class schema;

compiling the source code file to generate an executable file that implements methods of the class;

using one method of the class to construct the second data object; and

including the attribute values from the first data object into the second data object.

28. (Previously Presented) The system of claim 26, wherein the class schema includes for each attribute a name, data type and length of the attribute.

29. (Previously Presented) The system of claim 25, further comprising:

means for receiving a third data object implemented in the second programming language; and

means for adding the third data object to the database.

30. (Currently Amended) A system for managing database requests from an application that processes data objects in a first programming language, comprising:

a computer readable medium including a database having data objects implemented in a second programming language;

means for receiving a request from the application for ~~at least one~~ a data object in the database storing the data object in a second programming language, the data object having attributes and attribute values of a class;

means for accessing ~~each~~ the requested data object from the database;

means for transforming ~~each~~ the accessed data object to ~~one~~ a transformed data object implemented in the first programming language, wherein ~~each~~ the transformed data object includes the attributes and attribute values of the class in each accessed data object; and

means for returning, in response to the request, ~~each~~ the transformed data object ~~in the first programming language~~ to the application ~~that initiated the request~~.

31. (Currently Amended) The system of claim 30, wherein the means for transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object further performs for ~~each~~ the requested data object:

using a GET interface in the second programming language to access the attribute values in the accessed data object; and

using a SET interface in the first programming language to add each accessed attribute value from the accessed data object to the transformed data object.

32. (Previously Presented) The system of claim 30, wherein the application that processes data objects in the first programming language comprises a first application, further comprising:

means for receiving a request for at least one data object in the database from a second application that processes data objects in the second programming language;

means for accessing each requested data object from the database; and

means for returning each data object accessed from the database in response to the request from the second application to the second application.

33. (Currently Amended) The system of claim 30, further comprising:

means for providing ~~at least one~~ a class schema, wherein ~~each~~ the class schema includes information on one class and attributes of the class of ~~at least one~~ the data object in the database, wherein the means for transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object further performs, for ~~each~~ the accessed data object, using information on the attributes in the class schema for the class of the accessed data object to transform the accessed data object to the transformed data object.

34. (Currently Amended) The system of claim 30, wherein the application requesting the ~~at least one~~ data object processes data objects in one of the first programming language or a third programming language, further comprising:

means for determining whether the application requesting the ~~at least one~~ object processes data objects in the first programming language or the third programming language, wherein the accessed data object is transformed to the transformed data object implemented in the first programming language if the application requesting the at least one data object processes data objects in the first programming language;

means for transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object implemented in the third programming language if the application requesting the ~~at least one~~ data object processes data objects in the third programming language; and

means for returning, in response to the request, ~~each~~ the transformed data object ~~in the third programming language~~ to the application ~~that initiated the request~~.

35. (Currently Amended) A system for providing information on a class, comprising:

a computer readable medium;

means for receiving a definition of a class and attributes in the class of a first data object implemented in a first programming language;

means for generating a file in the computer readable medium; and

means for adding information to the file to provide a class schema representing ~~on~~ the class and each attribute in the ~~received~~ class. ~~definition to the generated file.~~

36. (Currently Amended) The system of claim 35, comprising

means to provide the file with a request to take an action on the first data object, the action associated with a second programming language; and

means to establish, using the class schema from the file, a second data object implemented in the second programming language to represent the first data object. ~~wherein the means for adding information on each attribute to the generated file further performs:~~

~~adding information on a name, length and data type of each attribute in the class in the received class definition to the generated file.~~

37. (Currently Amended) An article of manufacture including code for maintaining a database of data objects, wherein the code causes operations to be performed in an electronic device comprising:

receiving a request to store a first data object implemented in a first programming language in a database capable of storing data objects in a second programming language, the first data object including attributes and attribute values for a class;

transforming the first data object to a second data object implemented in a second programming language, wherein the second data object includes the attributes and attribute values of the class included in the first data object; and

~~adding~~ storing, in response to the request, the second data object to the database.~~, wherein the database is capable of storing multiple data objects implemented in the second programming language.~~

38. (Previously Presented) The article of manufacture of claim 37, further comprising:

receiving a class schema including information on the class and attributes of the first data object; and

using the received class schema to transform the first data object to the second data object.

39. (Previously Presented) The article of manufacture of claim 38, wherein using the received class schema to transform the first data object to the second data object further comprises:

generating a source code file in the second programming language to implement the class and attributes included in the class schema;

compiling the source code file to generate an executable file that implements methods of the class;

using one method of the class to construct the second data object; and

including the attribute values from the first data object into the second data object.

40. (Previously Presented) The article of manufacture of claim 39, wherein generating the source code file in the second programming language further comprises:

generating statements into the source code file to define SET and GET interfaces for each attribute in the class.

41. (Previously Presented) The article of manufacture of claim 40, wherein including the attribute values from the first data object into the second data object further comprises:

using at least one GET method in the first programming language to access the attribute values from the first data object; and using at least one SET method in the second programming language to set each attribute in the second data object to the corresponding accessed attribute value.

42. (Previously Presented) The article of manufacture of claim 40, wherein including the attribute values from the first data object into the second data object further comprises:

using at least one GET method in the first programming language to access the attribute values from the first data object; and

generating statements into the source code file to set the attributes in the second data object to the accessed attribute values from the first data object, wherein compiling the source code file produces the second data object with the attribute values set to the attribute values accessed from the first data object.

43. (Previously Presented) The article of manufacture of claim 38, wherein the class schema includes for each attribute a name, data type and length of the attribute.

44. (Previously Presented) The article of manufacture of claim 38, wherein the class schema is implemented in an Extensible Markup Language (XML) file.

45. (Previously Presented) The article of manufacture of claim 37, wherein the database comprises an object oriented database.

46. (Previously Presented) The article of manufacture of claim 37, wherein the first programming language comprises a non-Java object oriented language and wherein the second programming language comprises the Java programming language.

47. (Previously Presented) The article of manufacture of claim 37, further comprising:

receiving a third data object implemented in the second programming language; and

adding the third data object to the database.

48. (Currently Amended) The article of manufacture of claim 37, further comprising:

receiving a request to a store a third data object implemented in a third programming language including attributes and attribute values for one class;

transforming the third data object to a fourth data object implemented in the second programming language, wherein the fourth data object includes the attributes and attribute values of the class included in the third data object; and

~~adding~~ storing, in response to the request, the fourth data object to the database.

49. (Currently Amended) An article of manufacture including code for returning data objects from a database to an application that processes data objects in a first programming language, wherein the code causes operations to be performed in an electronic device comprising:

receiving a request from the application for ~~at least one~~ a data object in the database storing the data object in a second programming language, the data object having attributes and attribute values of a class;

accessing ~~each~~ the requested data object from the database, ~~wherein data objects in the database are implemented in a second programming language~~;

transforming ~~each~~ the accessed data object to ~~one~~ a transformed data object implemented in the first programming language, wherein ~~each~~ the transformed data object includes the attributes and attribute values of the class ~~in each~~ of the accessed data object; and

returning, in response to the request, ~~each~~ the transformed data object ~~in the first programming language~~ to the application ~~that initiated the request~~.

50. (Currently Amended) The article of manufacture of claim 49, wherein transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object further comprises for ~~each~~ the requested data object:

using a GET interface in the second programming language to access the attribute values in the accessed data object; and

using a SET interface in the first programming language to add each accessed attribute value from the accessed data object to the transformed data object.

51. (Previously Presented) The article of manufacture of claim 49, wherein the application that processes data objects in the first programming language comprises a first application, further comprising:

receiving a request for at least one data object in the database from a second application that processes data objects in the second programming language;

accessing each requested data object from the database; and

returning each data object accessed from the database in response to the request from the second application to the second application.

52. (Currently Amended) The article of manufacture of claim 49, further comprising:

providing ~~at least one~~ a class schema, wherein ~~each~~ the class schema includes information on one class and attributes of the class of ~~at least one~~ the data object in the database, wherein transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object further comprises, for ~~each~~ the accessed data object, using information on the attributes in the class schema for the class of the accessed data object to transform the accessed data object to the transformed data object.

53. (Currently Amended) The article of manufacture of claim 52, wherein ~~each~~ the class schema includes a length of each attribute in the class, and wherein using the information on the attributes in the class schema to transform ~~each~~ the accessed data object to ~~one~~ the transformed data object further comprises:

accessing information on the length for each attribute in the class schema to generate the transformed data object ~~to have~~ with a size at least equal to ~~the~~ a sum of lengths of all of the attributes in the class.

54. (Currently Amended) The article of manufacture of claim 49, wherein the application requesting the ~~at least one~~ data object is capable of processing data objects in one of the first programming language or a third programming language, further comprising:

determining whether the application requesting the ~~at least one~~ data object processes data objects in the first programming language or the third programming language, wherein the step of transforming ~~each~~ the accessed data object to the transformed data object implemented in the first programming language occurs if the application requesting the ~~at least one~~ data object processes data objects in the first programming language;

transforming ~~each~~ the accessed data object to ~~one~~ the transformed data object implemented in the third programming language if the application requesting the ~~at least one~~ data object processes data objects in the third programming language; and

returning ~~each~~ the transformed data object in the third programming language to the application that initiated the request.

55. (Currently Amended) An article of manufacture including code for providing information on a class, wherein the code causes operations to be performed in an electronic device comprising:

receiving a definition of a class and attributes in the class of a first data object implemented in a first programming language;

generating a file; and

adding information to the file to provide a class schema representing ~~on~~ the class and each attribute in the class. ~~received class definition to the generated file.~~

56. (Currently Amended) The article of manufacture of claim 55, comprising ~~wherein adding information on each attribute to the generated file further comprises:~~

~~adding information on a name, length and data type of each attribute in the class in the received class definition to the generated file.~~

providing the file with a request to take an action on the first data object, the action associated with a second programming language; and establishing, using the class schema from the file, a second data object implemented in the second programming language to represent the first data object.

57. (Currently Amended) The article of manufacture of claim 55, further comprising:

generating at least one tagged element into the file ~~including information on each~~ to represent at least one attribute in the class.

58. (Currently Amended) The article of manufacture of claim 57, wherein generating the at least one tagged element into the file for each attribute in the class further comprises for each attribute of the class:

generating one tagged element into the file including information on one of a name, a length, and a data type of the attribute. ~~[[;]]~~

~~generating one tagged element into the file including information on a length of the attribute; and~~

~~generating one tagged element into the file including information on a data type of the attribute.~~

59. (Previously Presented) The article of manufacture of claim 57, wherein the-generated file comprises an Extensible Markup Language (XML) file.

60. (Previously Presented) The article of manufacture of claim 55, further comprising: accessing the definition of the class, including information on attributes of the class, from a source code file of the class.